멀티미디어 전송을 위한 온라인 대역폭 평활화 기법

(Online Bandwidth Smoothing for Multimedia Transmission)

김 재 욱 † 하 란 †† 차 호 정 †††
(Jae-wook Kim) (Rhan Ha) (Hojung Cha)

요 약 네트워킹 환경에서 만족스러운 멀티미디어 응용 서비스를 제공하기 위해서는 효과적인 미디어 전송 구조가 필요하다. 그러나 인터넷 같은 일반적인 네트워크는 안정적인 멀티미디어 서비스를 위한 보장된 네트워크 대역폭을 제공하지 않는다. 따라서, 종단 시스템에서 멀티미디어 서비스를 지원하기 위한 일반적인 방법으로 버퍼를 사용한다. 기존의 제안된 버퍼 관리 기법은 크게 두 가지 영역으로 분류된다. 하나는 변화하는 네트워크 환경에 적응하기 위한 기법이고 다른 하나는 요구 대역폭을 평활화하는 기법이다. 그러나 전자는 서비스 품질의 심각한 손실을 일으킬 수 있고 후자는 동적인 네트워크 환경에 적응할 수 없다는 단점이 있다. 본 논문에서는 동적인 네트워크에 적응하며 대역폭 평활화를 통해 더 좋은 서비스 품질을 제공하는 적응성 있는 대역폭 평활화 기법을 제안한다. 또한 저장된 MPEG 비디오를 이용한 실험을 통해 제안된 기법의 사용으로 비디오 전송 품질이 향상됨을 보인다.

Abstract To deliver multimedia data, the periodic real-time delivery and large network bandwidth are required. However, common networks such as Internet cannot guarantee these requirements. Thus, in order to provide satisfactory multimedia application service, the mechanism which can transmit data effectively even with dynamic network bandwidth change is required. The efficient buffer management technique is a solution for the delivery of multimedia data on common networks. These buffer management techniques are classified into two approaches: one is adapting the changes in network load and the other is smoothing the bandwidth requirement. The former may occur the serious loss of service quality and the latter cannot be adapted to the dynamic network condition. In this paper we propose a bandwidth adaptive smoothing which is adaptive to the dynamic networks and also supports better quality of service by smoothing the bandwidth requirement. Simulation results with a prerecorded MPEG video show that the quality of video delivered is improved with the proposed technique.

1. 서 론

컴퓨터와 네트워크의 빠른 발전으로 원격지 사용자에게 다양한 멀티미디어 서비스가 가능하게 되었다. 화상회의나 주문형 비디오 같은 멀티미디어 응용은 주기적인 전송을 위한 큰 대역폭을 요구한다. 그러나 인터넷 같이 TCP/IP를 기반으로 하는 일반적인 네트워크는 주기적인 전송이나 요구 대역폭 할당을 보장하지 못한다. 따라서, 동적인 네트워크 환경에서 멀티미디어 트래픽의 효과적인 전송을 위한 접근 방법으로 버퍼를 사용하였다.여러 가지 측면의 버퍼 관리에 대한 연구들이 행하여졌는데, 이러한 연구들은 크게 동적 품질 제어(dynamic quality adjustment)와 정적 평활화(static smoothing)로 나누어진다. 동적 품질 제어 (이하 DQA) 기법은 네트워크의 가용 대역에 맞게 데이타의 품질을 제어하여 네트워크 부하의 변화에 적응한다[1, 2, 3]. 반면, 정적 평활화 (이하 SS) 기법은 각각의 데이타 집중에 앞서 클라이언트 버퍼로 데이타를 선적재하는 방법으로 집중

• 본 연구는 정보통신연구진흥원(과제번호: C1-1999-1234-00), 한국과학재단(과제번호: KOSEF97-0102-05-01-3, KOSEF97-01-00-12-01-5), 교육부 BK(과제번호: 99103101)의 후원으로 연구되었음.
† 학생회원 : 홍익대학교 정보공학과
hermess@unitel.co.kr
†† 종신회원 : 홍익대학교 컴퓨터공학과 교수
rhanha@cs.hongik.ac.kr
††† 종신회원 : 광운대학교 컴퓨터과학과 교수
hojungc@cs.kwangwoon.ac.kr
논문접수 : 2000년 2월 7일
심사완료 : 2000년 6월 2일

된 데이타를 분산시켜 멀티미디어 데이타의 요구 대역을 평활화한다[4, 5].

DQA는 일반적으로 온라인 응용에 사용된다. 이 기법은 동적 네트워크의 상태에 적응하기 위해 네트워크를 모니터링하고 그에 반응하여 가용 대역폭에 맞게 데이타의 품질을 제어한다. DQA는 네트워크의 지터 제거를 위해 버퍼를 사용하였을 뿐, 안정적인 서비스를 위한 요구 대역폭 평활화를 위해 사용하지는 않았다. VIC[1], continuous media toolkit[2], quarsar's adaptive streaming video player[3]이 DQA의 좋은 예이다. 이러한 시스템은 가용 네트워크 자원에 따라 요구 대역폭을 감소시키기 위해 데이타의 품질을 결정하는 요소들을 빈번하게 제어함으로서 심각한 서비스 품질의 손실을 일으킬 수 있다. 반면 SS는 오프라인 접근 방법이다. 이 기법은 대역폭 평활화를 통해 불필요한 서비스 품질의 손실을 줄인다. SS는 클라이언트 버퍼의 크기나 저장된 멀티미디어 데이타로부터 얻을 수 있는 정보를 분석하여 프레임율의 변화를 최소화하도록 전송계획을 수립한다. Critical bandwidth allocation[4]과 optimal smoothing algorithm[5]가 SS의 예이다. 그러나 SS는 네트워크의 지터에 대응하지 못한다. 클라이언트로의 전송과 재생 이전에 전송 계획을 수립하기 때문에 네트워크의 지터가 발생하게 되면 버퍼의 언더플로우를 초래하여 클라이언트의 재생이 불가능해질 수도 있다.

본 논문에서는 동적 네트워크에 적응하면서 요구 대역폭의 평활화를 통해 더 나은 서비스 품질을 제공하는 적응성 있는 온라인 대역폭 평활화 기법(bandwidth adaptive smoothing, 이하 BAS)을 제안한다. 제안된 BAS는 기존의 기법들과 다른 몇 가지 특징을 가지고 있다. 첫째, BAS는 동적 윈도우를 사용한다. 기존의 윈도우 기반의 평활화 기법[4, 6]은 네트워크 상태를 고려하지 않고 고정된 크기의 윈도우를 사용하기 때문에 네트워크 부하의 변화에 적응하지 못한다. 반면 BAS는 네트워크의 상태에 따라 동적 윈도우의 크기를 결정한다. 둘째, BAS의 클라이언트 버퍼는 네트워크 지터의 적응성뿐만 아니라 요구 대역의 평활화를 위해 사용한다. 클라이언트 버퍼는 두 영역으로 나누어, 한 부분은 DQA와 같이 네트워크 지터에 대비하여 선적재를 위해 사용하고 나머지 부분은 SS와 같이 대역폭 평활화를 위해 사용한다. 만일 전송 계획에 따라 데이타를 전송하는 과정에 네트워크의 지터가 발생하면, BAS는 지터에 의한 영향을 막기 위해 다른 부분에 선적재한 데이타를 소모한다. 마지막으로 BAS는 온라인 상에서 전송 계획을 수립하기 위하여 피드백 정보를 사용한다. 기존의 SS는 오프라인에서 전송 계획을 수립하기 때문에 피드백 정보를 필요로 하지 않을 뿐만 아니라 동적 네트워크 환경을 전송 계획에 반영하지 못한다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 BAS 기법의 원리와 알고리즘에 대해 기술하고 3장에서는 실험 결과를 분석하며 마지막으로 4장에서는 결론과 향후 연구과제에 대해 논한다.

2. Bandwidth Adaptive Smoothing 기법

BAS시스템은 그림 1에서와 같이 서버 측의 동적 윈도우와 클라이언트 쪽의 동적 버퍼로 구성되어 있다. 먼저 온라인 상태에서 유효한 전송 계획을 수립하기 위해 필요한 평활화 제약을 유도하고 이 제약 조건에서 동적 윈도우를 이용한 대역폭 평활화를 통해 전송 계획의 수립 과정과 네트워크 상태에 따른 동적 버퍼의 영역 변

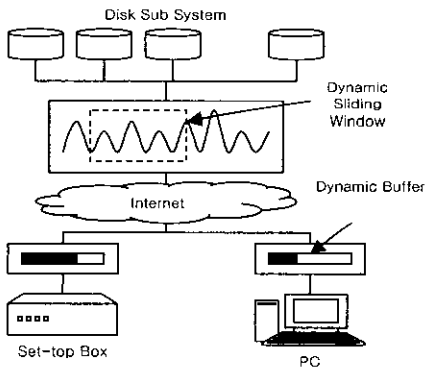


그림 1 BAS 시스템 모델

표 1 기호 정리

W	: 동적 윈도우의 시간 간격 크기
a	: 윈도우 슬라이딩 폭
Δ	: 윈도우 크기의 증가율
$L(t)$	: 클라이언트의 최소 데이타 소모량
$U(t)$	: 클라이언트의 최대 적재량
D	: 누적 전송 벡터
A	: 누적 적재 벡터
S_i	: 전송 계획
s	: 서버의 전송 속도
δ	: 서버의 전송 속도 증가율
B_c	: 클라이언트의 버퍼 크기
B_s	: 클라이언트 버퍼의 평활화 영역 크기
B_j	: 클라이언트 버퍼의 지터 영역 크기
θ	: 평활화 영역의 최대 증가량

화에 대하여 설명한다. 마지막으로 예제를 통하여 주어진 알고리즘에 따른 시스템의 동작을 살펴본다. 표 1은 알고리즘 설명을 위한 기호를 정의하였다.

2.1 온라인 평활화 제약

BAS 서버는 데이타의 집중에 앞서 클라이언트 버퍼에 데이타를 선적재하는 방법으로 저장된 멀티미디어 데이타의 요구 대역을 감소시킨다. 서버는 평활화에 앞서 프레임 사이즈 등의 멀티미디어 데이타에 대한 사전 정보와 클라이언트 버퍼의 크기에 기초하여 전송할 데이타 양의 상한과 하한 제약을 계산한다. 그리고, 이 두 제약 조건을 기초로 하여 데이타의 집중을 최소화하도록 전송 계획을 수립한다[5, 6, 7]. $i = 1, 2, \ldots, N$인 프레임 i가 f_i 바이트이고 N 프레임으로 구성된 비디오 스트림이 있다고 가정하자. 전체 스트림은 서버에 저장되어 있고 네트워크를 통해 클라이언트 버퍼로 전송된다. 각 시간 단위는 30 프레임의 풀-모션 비디오에 대응하는 1/30초로 가정한다. 이때, 클라이언트로의 계속적인 전송을 위해서 서버는 식 (1)에서 보듯이 시간 t, $t = 0, 1, \ldots, N$ ($L(0) = 0$) 안에 클라이언트 버퍼의 언더플로우를 막도록 $L(t)$이상 전송하여야 한다.

$$L(t) = \sum_{i=1}^{t} f_i \tag{1}$$

이것은 $L(0) = 0$이고 $t = 1, 2, \ldots, N$인 t까지 소모된 데이타의 양을 나타낸다. 비슷한 방법으로 클라이언트 버퍼의 오버플로우로 인한 데이타의 손실을 막기 위해 클라이언트는 식 (2)에서 보듯이 $U(t)$ 이상을 받아서는 안된다.

$$U(t) = L(t-1) + B_s \tag{2}$$

따라서 데이타의 손실 없는 재생을 위해서 전송 계획 S_i는 식 (3)과 같이 $U(t)$와 $L(t)$사이에 머물어야 한다.

$$L(t) \le S_i \le U(t) \tag{3}$$

그러나 제안된 BAS는 온라인 상태의 평활화 기법이다. 오프라인 평활화 기법의 경우 네트워크 대역에 대한 고려 없이 버퍼의 상태와 전체 멀티미디어 데이타의 정보를 이용하여 최적의 전송 계획을 만들지만 온라인 상태의 평활화 기법은 버퍼의 상태와 멀티미디어 데이타의 정보뿐만 아니라 네트워크 상태에 대한 고려가 필요하다. 하지만 네트워크의 상태를 예측할 수 없기 때문에 사전에 전체 멀티미디어 데이타의 정보를 이용하여 전송 계획을 수립할 수 없다.

따라서 온라인 상태에서의 멀티미디어 서버는 단지 제한된 정보를 가지고 데이타의 일부분만을 접근한다. 그러므로 BAS에서 전송 계획을 수립하기 위해서는 위에 제시된 두 평활화 제약 조건을 온라인 상태에 맞게 확장시켜야 한다.

정리 1. *$\tau < t$ 인 어떤시간 τ에서 t 사이의 온라인 평활화 제약은 $L'(\tau, t) = D_t - D_\tau$와 $U'(\tau, t) = L'(\tau, t-1) + B_s$로 확장되고 유효한 전송 계획은 $L'(\tau, t) \le S'(\tau, t) \le U'(\tau, t)$를 만족하여야 한다.*

증명: W 크기의 윈도우와 전송 계획 계산을 위한 지연 시간을 고려하자. 연속적인 재생을 위해 클라이언트에 적재되어야 할 데이타의 양을 나타내는 누적 적재 벡터 $A = \{A_0, A_1, \ldots, A_W\}$, 전송될 데이타의 양을 나타내는 누적 전송 벡터 $D = \{D_0, D_1, \ldots, D_W\}$라고 하고, W 시간 간격을 가진 윈도우의 전송 계획 계산을 위한 지연 시간을 $d(W)$로 정의한다. 이 때, 다음에 올 W' 시간 간격을 가진 윈도우의 전송 계획 수립을 위해서 서버는 $W - d(W')$ 이전에 W 구간의 재생을 위해 필요한 모든 데이타의 전송을 끝내야 한다. 그러므로 D_i와 A_i를 각각 시간 i에서 전송되어야 할 데이타의 누적 전송량과 재생에 필요한 누적 적재량이라고 하면 $A_W \le D_{W-d(W')}$가 되어야 하므로 시간 $0 \le i \le W - d(W')$ 일 때, $D_i = max(A_W / (W - d(W')) \times i, A_i)$가 되고 $W - d(W') < i \le W$ 일 때 $D_i = A_W$가 될 것이다. 따라서, 이를 이용하여 $\tau < t$ 일 때, 어떤 시간 τ부터 t까지의 새로운 온라인 상태 평활화 제약은,

$$L'(\tau, t) = D_t - D_\tau \tag{4}$$

와

$$U'(\tau, t) = L'(\tau, t-1) + B_s \tag{5}$$

가 되고 $\tau < t$인, 어떤 시간 τ에서 t 까지의 유효한 온라인 전송 계획 $S(\tau, t)$는 (6)과 같은 범위에서 구할 수 있다.

$$L'(\tau, t) \le S'(\tau, t) \le U'(\tau, t) \tag{6}$$

□

2.2 동적 슬라이딩 윈도우를 이용한 평활화

서버의 평활화 윈도우는 $\alpha < W$인 α만큼 슬라이딩하며 중첩되게 평활화를 통한 전송 계획을 수립한다. 이 알고리즘은 매 α마다 전송 계획을 수립하기 때문에 슬라이딩하지 않는 일반 윈도우에 비해 평균 W / α배 빈번한 전송 계획 수립이 필요하다. 그러나 빈번한 계산으로 네트워크의 변화를 빠르게 반영하여 적응성 있는 전송 계획을 작성할 수 있으며, 윈도우간의 경계를 넘은 데이타의 선적재로 더 좋은 평활화 효과를 나타낸다. 슬라이딩 폭인 α는 실험에 의해 가장 효율적으로 밝혀진 $W / 2$의 값을 사용한다[6, 7].

기존에 제안된 슬라이딩 윈도우는 네트워크의 상태에

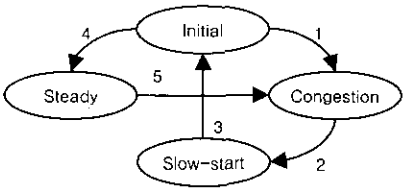


그림 2 BAS 상태 변화도

관계없이 크기가 고정되어 있었으나 BAS의 동적 슬라이딩 윈도우는 기존의 접근 방법과는 다르게 네트워크의 상태에 따라 크기가 변화한다. BAS 알고리즘은 클라이언트로부터 피드백 정보를 받아 네트워크의 상태를 예측하는데 확장된 TCP[8]나 SCP(Streaming Control Protocol)[9]와 같이 네트워크의 상태를 initial state, congestion state, slow-start state, steady state로 정의하고 각 상태에 따라 윈도우의 크기를 결정한 후, 그에 맞는 전송 계획을 수립한다. 정의된 네트워크 상태 전이는 그림 2에 나타내었다. initial state에서 서버는 윈도우 크기 W와 전송 속도 s를 초기 입력 값으로 설정한다. congestion state는 클라이언트 버퍼의 적재량이 네트워크의 상태 변화나 지터에 의해 B_l이하로 떨어졌을 때, initial state나 steady state로부터 시작된다(그림 2의 전이 1과 5). congestion state에서는 윈도우의 크기 W는 $W / 2$로 줄이고 전송 속도 $s = s + s \times \delta$로 설정한다. 윈도우의 크기가 반으로 줄기 때문에 전송 계획이 빈번하게 수립되고 전송 속도가 증가함에 따라 버퍼로의 재적재가 가능하다. congestion state에서 서버는 단지 윈도우의 크기와 전송 속도만을 바꾸고 바로 다음의 slow-start state로 넘어간다(그림 2의 전이 2). slow-start state에서 서버는 윈도우 크기 W와 전송 속도 s를 초기 입력 값과 같은 상태로 만들기 위한 시도를 한다. 서버는 클라이언트 버퍼의 평활화 영역 크기 B_s의 증가율에 맞춰 서서히 윈도우 크기 W를 증가시킨다. 만일 윈도우의 크기 W가 초기 입력 값과 같아지면 slow-start state를 마치고 initial state로 넘어간다(그림 2의 전이 3). 윈도우의 증가율 Δ는 멀티미디어 데이타의 특성에 따라결정한다. steady state는 네트워크의 상태가 안정적인 경우 초기 상태로부터 시작된다(그림 2의 전이 4). 이 상태에서는 더 넓은 영역을 대상으로 평활화를 시도하여 향상된 서비스 품질을 얻기 위해 윈도우의 크기를 증가시킨다. 최대 윈도우 증가 크기는 클라이언트 버퍼의 평활화 영역의 최대 증가 크기 θ에 의해 제한된다. 구체적인 알고리즘은 표 2에 나타내었다.

여기서 동적 윈도우 크기의 변화에 따라 슬라이딩 시간 간격이 바뀌어 전송 계획의 계산 주기가 바뀌게 된다. 그러나 이에 따른 오버헤드는 변함이 없다.

정리 2. *N 프레임 범위를 가진 윈도우 W를 이용하여 전송 계획을 계산하는데 따른 오버헤드를 C라고 할 때, 동적 윈도우 크기의 변화에 따라 전송 계획의 계산 주기가 변하여도 전체적인 오버헤드는 변하지 않는다.*

증명: 전송 계획을 계산하는 오버헤드 C는 윈도우 내의 프레임 수에 비례한다. 만일 서버가 congestion state로 들어가면 윈도우의 범위가 $N / 2$ 감소하여 전송 계획의 계산 횟수가 2배로 증가하게 된다. 그러나 윈도우 내의 프레임 범위가 반으로 감소하기 때문에 오버헤드 역시 반으로 줄어들게 된다. 결과적으로 전체적인 오버헤드는 $C / 2 \times 2$배로 변함이 없다. 따라서 윈도우의 크기가 증가하거나 감소하더라도 계산 빈도와 프레임 수는 반비례하기 때문에 전체적인 오버헤드에는 변함이 없다. □

네트워크 상태에 따라 윈도우 크기가 결정되면 서버는 현재 가용 대역폭으로 동적 윈도우 내의 모든 데이타가 전송될 수 있는지 검사한다. 만일 전송이 불가능하다면 서버는 네트워크의 대역폭에 맞추기 위해 서비스 품질을 제어한다. 이렇게 서비스 품질이 결정된 데이타를 이용하여 서버는 유효한 전송 계획을 구한다. 유효한 전송 계획은 위에서 언급한 식 (6)을 만족하도록 만들어야 한다. 전송 계획을 만들기 위해서 서버는 식 (4)와 (5)에 제시된 L과 U를 위반하지 않는 가장 긴 단조 증가 직선을 찾는 방법으로 진행된다. 그림 3은 전송 계획 S_i를 만드는 예를 나타낸다. 처음 시작점 t =0에서 시작하고 가장 긴 증가 직선 1을 찾는다. 그러나 버퍼의 언더플로우를 초래하므로 가장 가까운 이전 점 a에서 전송률을 증가시켜야만 한다. 전송률의 증가를 최소화하기 위해 직선 2를 찾게 되지만 역시 버퍼의 오버플로우를 초래한다. 같은 이유로 다음 직선을 구한다. 결과적으로 구하고자하는 전송 계획 S_i는 그림 3의 아래 부분에 나

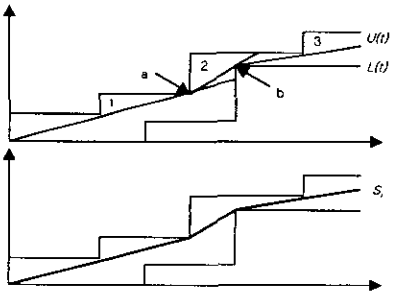


그림 3 전송 계획 S_i의 수립 예

표 2 동적 윈도우 변화 알고리즘

```
PROCEDURE Dynamic_Window
s' = s;W' = W;
REPEAT
status = read_feedback;
switch (status) {
case INITIAL :
W = initial_input_value;
if (s > s') s = s';// 전송 속도 초기화
break;
case CONGESTION :
W = W / 2;
s = s + s * δ;// 전송 속도 증가
break;
case SLOW_START :
W = W + Δ;
break;
case STEADY :
if (W < W' * (B_s + θ) / B_s)
W = W + W * (θ / B_s)
// W가 최대 값보다 작으면 W 증가
break;
}
UNTIL
END PROCEDURE
```

타나 있다. 예와 같이 BAS는 유효한 전송 계획의 수립을 위해 [5]에 제시된 $O(n)$의 복잡도를 가진 *optimal off- line smoothing algorithm*을 사용한다.

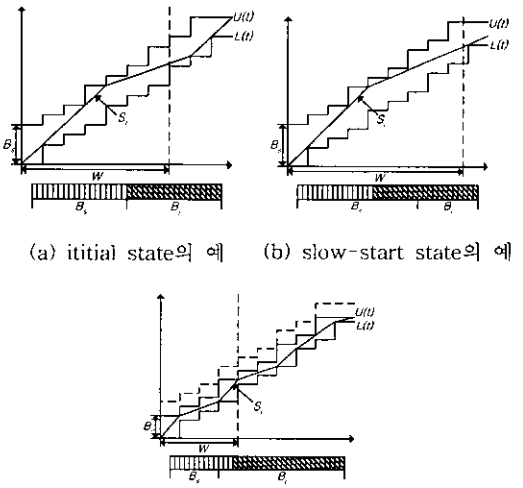
2.3 적응성 있는 클라이언트 버퍼링

제안된 BAS에서 클라이언트의 버퍼는 지터의 제거뿐만 아니라 멀티미디어 스트림의 요구 대역폭 평활화를 위해 사용된다. BAS는 고정된 B_c의 크기를 가진 클라이언트 버퍼를 사용하고 이 고정 크기 버퍼는 B_j 크기를 가진 지터 대비 영역과 B_s크기를 가진 평활화 대비 영역의 두 부분으로 나뉘어진다. 지터 영역은 발생될 지터에 대비하여 재생 시작 전에 보통의 데이타로 채워진다. 평활화 영역은 서버의 동적 윈도우와 함께 전송 계획을 만드는데 사용되고 결정된 전송 계획에 따라 평활화된 데이타의 선적재가 이루어진다. 만일 결정된 계획에 따라 전송하는 도중에 지터가 발생하여 전송에 차질이 생기면 BAS는 지터 대비를 위해 선적재하여 둔 데이타를 소모하게 된다. 지터 영역과 평활화 영역으로 나누어진 두 가상의 영역은 그림 2에 나타난 네트워크의 상태에 따라 그 크기가 변하게 된다.

초기상태에서 클라이언트 버퍼의 크기 B_c는 초기 입력 값으로 지정되고 각 영역의 크기 B_j와 B_s는 각각 B_c / 2를 갖게 된다. 이후, 서버는 지터 영역의 크기 B_j만큼 선적재 한 후, 평활화 영역과 서버의 동적 윈도우를 이용하여 결정한 전송 계획에 따라 데이타를 적재하고 소모하기 시작한다. 만일 네트워크의 상태 변화나 지터가 발생하지 않는다면 초기에 적재하여 둔 B_j 이상의 데이타는 버퍼에 유지될 것이다. 그러나 반대로 버퍼의 적재량이 초기에 적재하여 둔 B_j 이하로 떨어진다면 네트워크의 변화나 지터가 발생하였을 것으로 예측하고 서버에 상태 변화를 알린 후, congestion state로 들어간다. congestion state에서는 두 영역의 크기에 변화가 생긴다. 먼저 평활화 영역의 크기 B_s는 B_s / 2로 감소하고 지터 대비 선적재 영역의 새로운 크기 B_j는 다시 B_c - B_s가 된다. 서버의 윈도우의 크기가 줄어 선적재 데이타의 양이 줄어든 것에 비례하여 평활화 영역의 크기를 줄여 버퍼의 낭비를 줄인다. 또한 지터 대비 선적재 영역의 크기를 증가시킴으로서 적재량을 늘려 변화된 네트워크의 상태나 지터의 발생에 대비할 여지를 만들었다. congestion state에서는 버퍼의 각 영역 크기를 변화시킨 후, slow-start state로 들어가게 되고 초기 상태로 돌아가기 위한 준비를 한다. 평활화를 위한 영역의 크기가 서서히 증가하게 되고, 지터 대비 선적재 영역은 상대적으로 감소하게 된다. 평활화영역 크기의 증

표 3 동적 버퍼 변화 알고리즘

```
PROCEDURE DYNAMIC_BUFFER
REPEAT
status = Monitor(Client_Buffer);
feedback(status) {
switch(status) {
case INITIAL :
B_c = initial_input_value;
B_s = B_c / 2;
B_j = B_c - B_s;
break;
case CONGESTION :
B_s = B_s / 2;
B_j = B_c   B_s;
break;
case SLOW_START :
B_s = B_s + B_c * (Δ / W);
B_j = B_c - B_s;
break;
case STEADY :
if (B_s < B_s + θ) {
B_s = B_s + B_c * (Δ / W);
B_j = B_c   B_s;
// B_s가 최대 증가량이 아니면, B_s 증가
}
break;
}
UNTIL
END PROCEDURE
```



(a) ititial state의 예 (b) slow-start state의 예

(c) steady state의 예

그림 4 네트워크를 통한 전송의 예

가율은 윈도우 크기의 증가율에 비례하여 $B_c \times (\Delta / W)$ 씩 증가한다. 평활화 영역의 크기가 초기 상태와 같게 되면 slow-start state를 빠져 나오게 된다. 반대로 네트워크의 상태가 양호하여 steady state로 들어가게 되면 평활화 버퍼의 크기를 서서히 증가시켜 더 좋은 평활화 효과를 거두도록 하였다. 만일 지터 대비 선적재 영역의 크기가 상대적으로 줄어 적재량이 줄었을 때, 지터의 발생으로 인한 버퍼 전체의 언더플로우를 막기 위해 평활화 영역의 최대 증가 크기는 θ로 제한한다. 자세한 내용은 표 3의 알고리즘에 나타내었다.

2.4 예제

이 절에서는 예제를 통해 BAS의 알고리즘을 다시 살펴본다. 그림 4(a)는 네트워크 전송을 위한 초기 상태를 나타낸다. 초기 상태에서는 지터의 대비를 위해 B_f만큼의 데이타를 선적재하여 두고, W의 크기를 가진 서버의 동적 윈도우와 B_s 크기를 가진 클라이언트 버퍼의평활화 영역을 이용하여 구한 전송 계획에 따라 데이타를 클라이언트로 전송하여 적재하고 소모한다. 만일 네트워크의 변화나 지터에 의해 버퍼의 적재량이 B_f이하로 떨어진다면 congestion state로 들어가 이전 절에서 제시한 알고리즘에 의해 서버의 윈도우의 크기 및 클라이언트의 두 영역의 크기를 바꾼 후, slow-start state로 들어간다. 그림 4(b)는 congestion state로 진입 후, initial state로 돌아오기 위해 slow-start state에 있는 모습이다. 실선으로 나타나 있는 초기상태에 비해 B_s의 감소로 $L(t)$와 $U(t)$의 간격이 좁아지게 된다. 따라서, $L(t)$와 $U(t)$의 두 제약을 위반할 가능성이 커지게 되므로 넓은 영역으로의 평활화가 어렵게 된다. 버퍼로 선적재해야 할 데이타의 양은 평활화 영역의 크기 B_s의 감소로 인해 줄어들게 된다. 서버는 윈도우의 슬라이딩 폭 α가 작아지기 때문에 더 빈번하게 전송 계획을 계산하게 되므로 네트워크의 상태를 쉽게 반영할 수 있다. 마지막으로 그림 4(c)은 steady state를 나타낸다. steady state에서는 윈도우의 크기 W와 클라이언트 버퍼의 평활화 영역의 크기 B_s의 증가로 인해 더 넓은 영역의 평활화가 가능해 진다. W와 B_s의 최대 증가 크기는 이전 절에 언급한 표 2와 3의 알고리즘에 의해 제한된다.

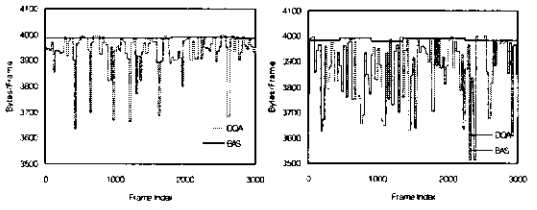
3. 실험 결과

이 절에서는 BAS를 사용하였을 때 얻어지는 성능을 평가하기 위한 실험 환경과 방법을 살펴보고, 그 결과를 분석 설명한다. BAS를 사용하여 향상되는 전송 데이타의 서비스 품질, 평활화 효과 및 네트워크의 적응성을 평가하기 위해 MPEG-1 비디오 파일을 이용한 시뮬레이션을 실시하였다. 시뮬레이터는 그림 1의 시스템 모델과 같이 디스크로부터 읽은 데이타를 서버의 동적 슬라이딩 윈도우를 통해 평활화하여 네트워크를 통해 클라이언트의 동적 버퍼로 전송되고 프레임 단위로 클라이언트 버퍼로부터 주기적으로 소모되는 구조를 가지고 있다. 본 실험에서는 BAS를 사용하여 향상되는 효과를 분석하기 위해 DQA의 기법과 비교하였다. SS는 오프라인 기반의 접근 방식이기 때문에 BAS와의 비교는 따로 하지 않는다. 본 실험에서 비교를 위해 사용하는 DQA기법은 다음과 같다. 서버는 먼저 디스크로부터 데이타를 읽어들여 클라이언트로 전송을 하고 전송 받은 데이타는 지터 발생에 대비하기 위하여 클라이언트 버퍼에 선적재하여 두고 B_f 이상 적재가 완료된 후 재생을 시작한다. 전송 중 지터가 발생하면 선적재하여 둔 데이타를 이용하여 지터를 제거한다. 또한 전송시 MPEG 비디오의 요구 대역을 네트워크의 가용 대역폭에 맞추기 위하여 프레임 드롭핑을 통해 서비스 품질을 제어한다. 프레임 드롭핑은 MPEG 파일의 프레임 종속성을 고려하여 B, P, I의 순서로 진행한다. 실험에 사용된 시뮬레이터는 C++을 이용하여 제작되었으며 윈도우 NT환경에서 작동하도록 하였다. 모든 실험은 10MB/sec의 Ethernet 환경에서 실시하였고 서버와 클라이언트는 동일한 서브넷에 위치하도록 하였으며 120KB/sec의 전송 대역과 각각 128KB의 클라이언트 버퍼를 사용하였다. 또한 네트워크의 변화나 지터 발생에 대한 적응성을 알아보기 위해 가상적으로 전체 전송의 1%만큼

균일 분포의 지연 지터를 주입하였다. BAS의 경우 서비스 품질의 제어를 위해 DQA와 같이 프레임 드롭핑을 사용하였으며 초기의 각 파라미터 값으로 윈도우 크기 W = 30초, 클라이언트 버퍼의 평활화 영역 최대 증가 크기 θ = 180, 윈도우 크기의 증가폭 Δ = 30, slow-start state에서 서버의 전송 속도 증가율 δ = 0.1로 사용하였다. 실험을 위한 데이타는 제안한 알고리즘이 요구 대역 분포에 관계없이 좋은 결과를 보이는지 알아보기 위해 그림 7, 그림 8과 표 4에서와 같이 서로 다른 요구 대역 분포를 가진 영화 '링 1'과 '링 2'를 사용하였다. 두 영화는 32,000여 프레임으로 구성되어 있으며 30프레임의 GOP와 352*240의 크기를 가진 I, B, P의 세 종류 프레임으로만 구성되어 있다.

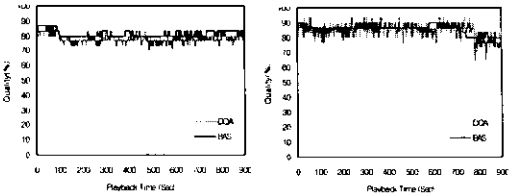
우선 DQA와 BAS의 평활화 효과를 비교하여 보았다. 좋은 평활화 효과는 데이타의 집중을 효과적으로 분산하여 불필요한 서비스 품질의 손실을 막아 안정적인 서비스를 제공할 수 있을 뿐만 아니라 네트워크의 활용도도 높일 수 있다.

그림 5에서 볼 수 있듯이, BAS는 전송률의 변화가 작은 반면 DQA의 전송률은 빈번하고 크게 변화한다. 그 이유는, DQA의 경우 네트워크의 가용 대역폭에만 초점을 두고 그에 맞게 서비스 품질을 제어하여 보내기 때문이다. 반면 BAS의 경우는 전송률의 큰 변화 없이 안정적인 서비스가 진행되고 있음을 볼 수 있다. 그러나, BAS가 그래프에서 보이는 DQA 전송률의 평균보다 더 높은 전송률로 서비스하고 있는데 그 이유는 다음과 같이 설명될 수 있다. DQA의 경우 가용대역폭에 맞춰 프레임 드롭핑을 통해 서비스 품질을 떨어뜨려 상대적으로 요구 대역을 낮춘 후 전송을 하기 때문에 전체적인 전송률이 낮은 반면, BAS의 경우는 평활화를 통해 데이타의 집중을 효과적으로 분산시켜 프레임 드롭이 상대적으로 일어나지 않았을 뿐만 아니라 버퍼로의 선적재를 위해 프레임 크기에 비해 더 많은 데이타를 전송하기 때문이다.



(a) '링 1' (b) '링 2'

그림 5 평활화 효과의 비교

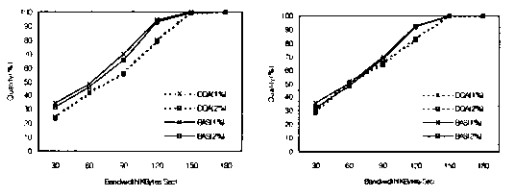


(a) '링 1' (b) '링 2'

그림 6 서비스 품질의 비교

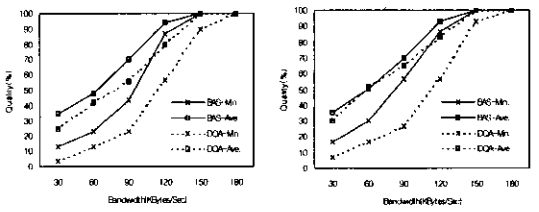
그림 6은 서비스 품질의 비교를 나타낸다. 여기서 서비스 품질은 30프레임 중 드롭되지 않고 서비스 된 프레임 수의 백분율로 정의한다. BAS의 경우는 앞에서 언급했듯이 사전에 평활화를 통해 전송 계획을 수립하기 때문에 불필요한 프레임의 드롭을 막아 DQA에 비해 안정적이며 더 좋은 서비스 품질로 서비스하고 있음을 알 수 있다. 그러나 때때로 DQA가 더 좋은 서비스 품질을 보인다. 이런 이유는 BAS의 경우 DQA와 같이 더 높은 품질로 서비스가 가능함에도 불구하고 전체적인 서비스 품질의 향상을 위해 일부 서비스 품질의 손실을 감수하고 전송 대역의 일부를 데이타의 선적재를 위해 할당하기 때문이다.

그림 7은 주입된 지터의 비율에 따른 서비스 품질의 변화를 나타낸다. 실험을 위해 전체 전송 중 1%, 2%의 지연 지터를 각각 주입하였다. 먼저 DQA의 경우 네트워크의 지터가 발생하여 버퍼의 적재 데이타가 소모되게 되면 재적재를 위해 전송 속도를 상향조정한다. 따라서 실제 데이타의 전송을 위한 대역폭이 상대적으로 줄어 서비스 품질의 제어가 일어나기 때문에 서비스 품질의 손상이 일어나지만 빠른 대처로 원래 품질에 비해 크게 손상되지 않음을 알 수 있다. 반면 BAS의 경우, 지터가 늘어나게 되면 윈도우 평활화 작업에 의해 DQA보다 반응시간이 느려 상대적으로 서비스 품질이 떨어짐을 알 수 있다. 그러나 윈도우의 크기가 좁아지고 전송 계획 수립 시간이 단축될 뿐만 아니라 평활화를 통해 불필요한 서비스 품질의 손실을 막기 때문에 여전히



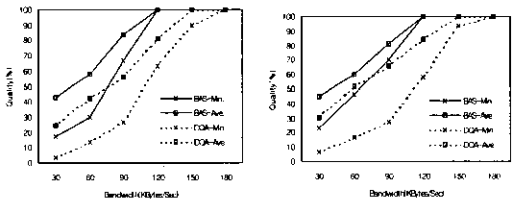
(a) '링 1' (b) '링 2'

그림 7 지터의 변화에 따른 서비스 품질의 비교



(a) '링 1' (b) '링 2'

그림 8 1MB 클라이언트 버퍼 사용시 최소 및 평균 서비스 품질의 비교



(a) '링 1' (b) '링 2'

그림 9 2MB 클라이언트 버퍼 사용시 최소 및 평균 서비스 품질의 비교

DQA에 비해 좋은 서비스 품질을 보임을 알 수 있다.

마지막으로 버퍼 크기의 변화에 따른 두 기법의 서비스 품질의 변화를 비교하여 보았다. 그림 8과 9에서는 지연 지터 1%에서 1MB와 2MB 클라이언트 버퍼를 사용하였을 때 DQA와 BAS를 통해 얻을 수 있는 평균(Ave.) 및 최소(Min.) 서비스 품질의 비교를 나타내었다. 그림의 결과에서 보듯이 BAS가 DQA에 비해 전체적으로 높은 서비스 품질을 보인다. '링 1'의 경우 BAS가 DQA보다 최소 서비스 품질의 경우 12. 9%, 평균 서비스 품질의 경우 13. 4%의 성능 향상을 보였으며, '링 2'의 경우 BAS는 DQA에 비해 평균 서비스 품질 6. 3%, 최소 서비스 품질 19. 2%의 성능 향상을 얻을 수 있었다. 또한 DQA는 네트워크의 전송 대역에 초점을 두고 서비스 품질을 제어하기 때문에 버퍼의 크기의 변화에 큰 영향을 받지 않음을 알 수 있으며 반면 BAS는 클라이언트 버퍼로의 선적재를 통해 평활화를 시도하기 때문에 버퍼의 크기가 증가하면 그에 대응하여 향상된 서비스 품질을 보인다. 한편 최초 화면이 보이기까지의 응답시간은, DQA의 경우 버퍼에 데이타를 선적재하여 모두 채우는데 걸리는 시간이 되고 BAS의 경우 버퍼의 절반을 선적재하여 채우는 시간과 첫번째 전송 계획 수립에 걸리는 시간의 합이 된다. 따라서 버퍼의 크기가 커질 경우 두 기법의 응답시간은 길어지게 된다. 그러나 BAS는 버퍼의 절반만 선적재 하며 동적 슬라이딩 윈도우의 크기가 작을 경우 전송 계획 수립에 짧은 시간이 소요되므로 DQA에 비해 좋은 응답시간을 기대할 수 있다.

4. 결 론

버퍼 관리 기법은 네트워크 환경에 있는 종단간 시스템에서 안정적인 멀티미디어 서비스를 지원하기 위한 한 방편으로 사용되어 왔다. 기존에 제안된 버퍼 관리 기법은 크게 네트워크의 적응성에 초점을 둔 DQA와 요구 대역폭 평활화를 통해 안정적인 서비스 품질에 초점을 둔 SS기법으로 나누어 볼 수 있다.

본 논문에서는 기존에 제안된 버퍼 관리 기법의 특징과 장단점에 관하여 살펴보았다. 동적 네트워크 상에서 멀티미디어 서비스 품질을 최대화하기 위해서 정적인 평활화 방법의 이점을 적용할 필요성을 발견하고 이를 이용하여 동적 네트워크 환경에 잘 적응하며 요구 대역의 평활화를 통해 더 좋은 서비스 품질을 보이는 적응성 있는 온라인 대역폭 평활화 기법을 제안하였다. 이 방식은 서버의 동적 슬라이딩 윈도우와 클라이언트의 동적 버퍼를 이용하여 실제 네트워크의 상태에 따라 평활화 영역의 범위를 변화시키며 불필요한 서비스 품질의 손실을 최소화하는 방법이다. 저장된 MPEG 비디오를 이용한 실험에서 BAS는 기존의 DQA의 기법보다 더 좋은 평활화 효과 및 서비스 품질을 나타낸다.

제안된 대역폭 평활화 기법에서 아직 고려되지 않은 부분은 윈도우 크기의 증가율(Δ), 전송률 증가량(δ), 평활화 영역의 최대 증가폭(θ)이다. 최적의 값을 구하는 방법은 연구 중에 있으며 적응성 있는 대역폭 평활화 기법을 이용한 멀티미디어 시스템을 구현과 전체적인 성능 평가는 향후 과제로 남긴다.

참 고 문 헌

[1] Steve McCanne and V. Jacobson, "VIC: A Flexible Framework for Packet Video," Proceedings of ACM Multimedia, Nov. 1995.

[2] L. A. Rowe, K. Patel, B. C. Smith and K. Liu, "MPEG Video in Software Representation, Transmission and Playback," Proceedings of IS&T/SPIE International Symposium on Electronic Imaging: Science and Technology, Feb. 1994.

[3] J. Walpole, R. Koster, S. Cen and D. Towsley, "A Player for Adaptive MPEG Video Streaming over The Internet," Proceedings of 26th Applied

Imagery Pattern Recognition Workshop, SPIE, Oct. 1997.
[4] W. Feng and S. Sechrest, "Smoothing and Buffering for Delivery of Prerecorded Compressed Video," Proceedings of IS&T/SPIE Multimedia Computing and Networking, Feb. 1995.
[5] J. Salehi, Z. Zhang, J. Kurose and D. Towsley, "Supporting Stored Video: Reducing Rate Variability and End-to-End Resource Requirements through Optimal Smoothing," Proceedings of ACM SIGMETRICS, May 1996.
[6] J. Rexford, S. Sen, J. Dey, W. Feng, J. Kurose, J Stankovic and D. Towsley, "Online Smoothing of Live, Variable-Bit-Rate Video," University of Massachusetts Computer Science Technical Report, Jul. 1998.
[7] S. Sen, J. Rexford, J dey, J. Kurose and D. Towsley, "Online Smoothing of Variable-Bit-Rate Streaming Video," Tech. Rep. , University of Massachusetts Computer Science Department, 1998.
[8] T. Lakshman and U. Madhow, "The Performance of TCP/IP for Network with High Bandwidth-Delay Products and Random Loss," IEEE/ACM Transactions on Networking, Vol. 5, No. 3, Jun. 1997.
[9] S. Cen, J. Walpole and C. Pu, "Flow and Congestion Control for Internet Media Streaming Applications. " Proceedings of SPIE Multimedia Computing and Networking), 1998.

김 재 욱
1998년 홍익대학교 컴퓨터공학과 졸업. 2000년 홍익대학교 대학원 정보공학과 실시간 시스템 전공 공학석사. 2000년 ~ 현재 유니텔 주식회사. 관심분야는 실시간 시스템, 멀티미디어 시스템.

하 란
1987년 서울대학교 컴퓨터공학과 졸업. 1989년 서울대학교 컴퓨터공학과 석사. 1989년 3월 ~ 1990년 7월 한국통신 전임연구원. 1995년 University of Illinois at Uubana-Champaign 전산학 박사. 1995년 9월 ~ 현재 홍익대학교 컴퓨터공학과 조교수. 관심분야는 실시간 시스템, 멀티미디어 시스템, 분산 시스템.

차 호 정
정보과학회논문지 : 정보통신
제 27 권 제 1 호 참조